

additional notes on survival analysis

Residuals

- Not straightforward to define residuals for binary outcomes
- One possibility: $r_i = \delta_i - \hat{H}(t_i)$
"martingale residual"

max=1 but can be arbitrarily negative

Fleming and Harrington (1991) showed that if:

$$h(t) = h_o(t)\phi(x^*)\exp(\beta^T X)$$


unknown function of a covariate

then:

$$E[R | X^*] \approx [\phi(X^*) - \bar{\phi}] \sum \frac{\delta_i}{n}$$

so, plots of the martingale residuals against individual covariates should be linear if the model is correct

Deviance Residuals

- Behave like residuals from ordinary linear regression
- Should be symmetrically distributed around 0 and have standard deviation of 1.0.
- Negative for observations with longer than expected observed survival times.
- Plot deviance residuals against covariates to look for unusual patterns.

Schoenfeld Residuals

- Schoenfeld (1982) proposed the first set of residuals for use with Cox regression packages
 - Schoenfeld D. Residuals for the proportional hazards regression model. *Biometrika*, 1982, 69(1):239-241.
- Instead of a single residual for each individual, there is a separate residual for each individual *for each covariate*
- Based on the individual contributions to the derivative of the log partial likelihood (see chapter 6 in Hosmer and Lemeshow for more math details, p.198-199)
- Note: Schoenfeld residuals are not defined for censored individuals.

Where K is the covariate of interest,

the Schoenfeld residual is the covariate-value, X_{ik} , for the person (i) who actually died at time t_i minus the expected value of the covariate for the risk set at t_i (=a weighted-average of the covariate, weighted by each individual's likelihood of dying at t_i).

$$\text{residual} = x_{ik} - \sum_{j \in R(t_i)} x_{kj} p_j$$

Plot Schoenfeld residuals against time to evaluate PH assumption

```
cox <- coxph(Surv(days,1-censor) ~ trmt,foo)
residuals(cox)
residuals(cox,type="scaledsch")
print(cox.zph(cox))
```

VA lung cancer data

- treatment: standard or test
- celltype: one of four types
- Karnofsky score: higher is healthier
- diagnosis: time since diagnosis
- age: in years
- therapy: prior therapy yes or no

```
(VA.cox <- coxph(Surv(stime,status) ~ treat + age + Karn  
+ diag.time + cell + prior, data=VA))
```

Call:

```
coxph(formula = Surv(stime, status) ~ treat + age + Karn + diag.time +  
      cell + prior, data = VA)
```

	coef	exp(coef)	se(coef)	z	p
treat2	2.95e-01	1.343	0.20755	1.4194	1.6e-01
age	-8.71e-03	0.991	0.00930	-0.9361	3.5e-01
Karn	-3.28e-02	0.968	0.00551	-5.9580	2.6e-09
diag.time	8.13e-05	1.000	0.00914	0.0089	9.9e-01
cell2	8.62e-01	2.367	0.27528	3.1297	1.7e-03
cell3	1.20e+00	3.307	0.30092	3.9747	7.0e-05
cell4	4.01e-01	1.494	0.28269	1.4196	1.6e-01
prior10	7.16e-02	1.074	0.23231	0.3082	7.6e-01

Likelihood ratio test=62.1 on 8 df, p=1.8e-10 n= 137

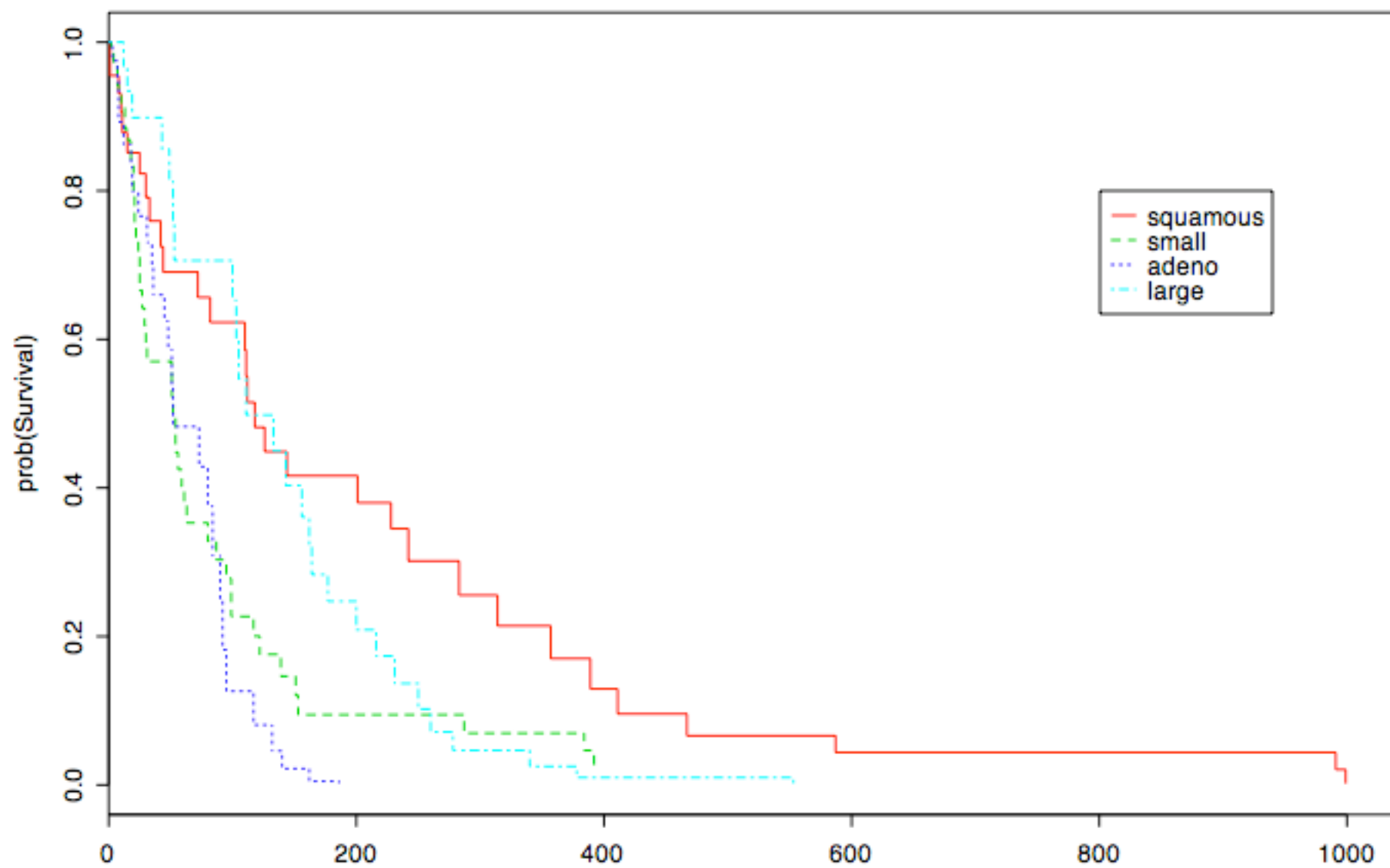
```
(VA.coxs <- coxph(Surv(stime,status) ~ treat + age +
Karn + diag.time + strata(cell) + prior, data=VA))
```

	coef	exp(coef)	se(coef)	z	p
treat2	0.28590	1.331	0.21001	1.361	1.7e-01
age	-0.01182	0.988	0.00985	-1.201	2.3e-01
Karn	-0.03826	0.962	0.00593	-6.450	1.1e-10
diag.time	-0.00344	0.997	0.00907	-0.379	7.0e-01
prior10	0.16907	1.184	0.23567	0.717	4.7e-01

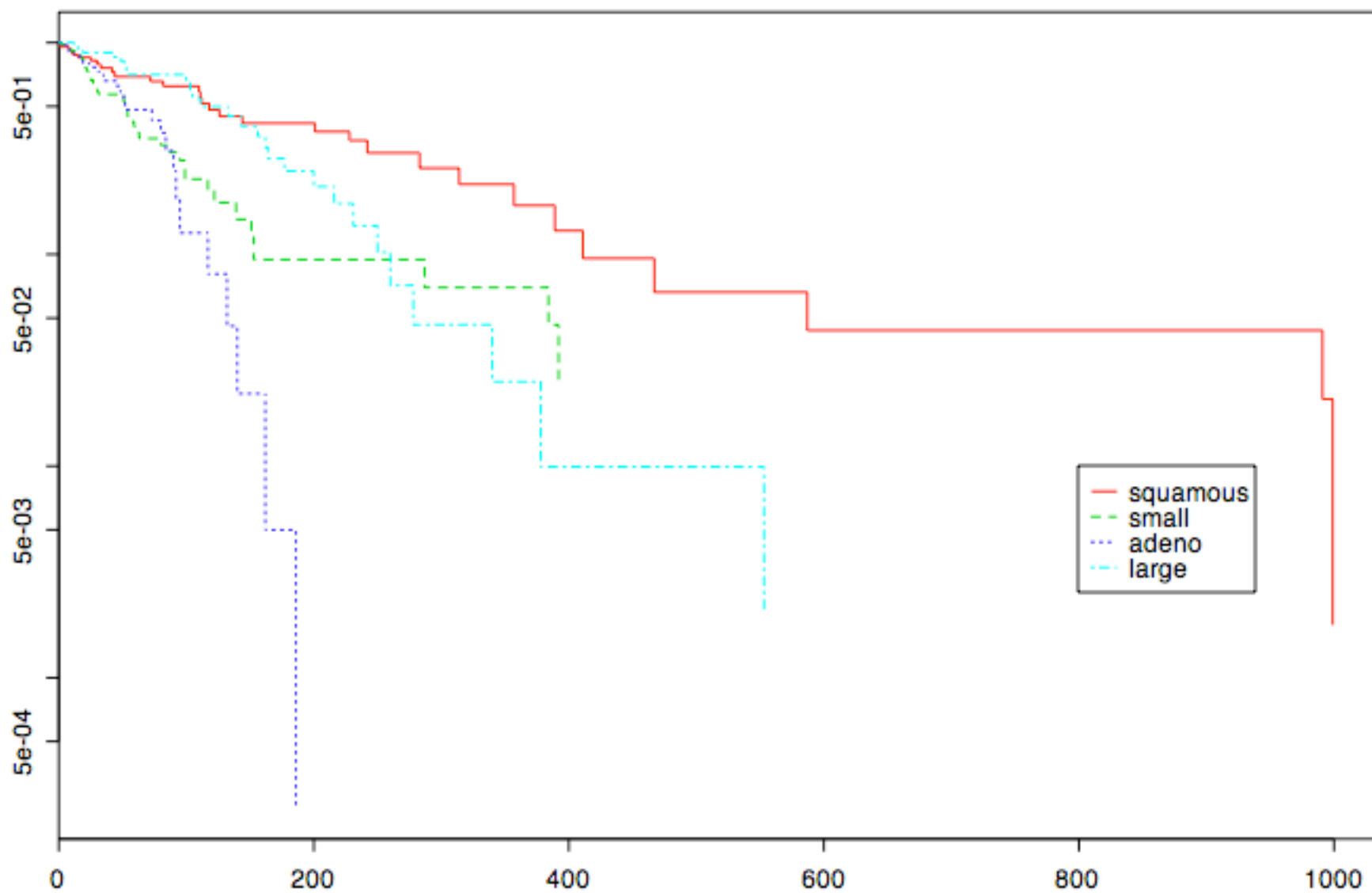
Likelihood ratio test=44.3 on 5 df, p=2.04e-08 n= 137

```
plot(survfit(VA.coxs),lty=1:4, col=2:5,ylab="prob(Survival)")
legend(800,0.8, c("squamous", "small", "adeno", "large"),
lty=1:4, col=2:5)
```

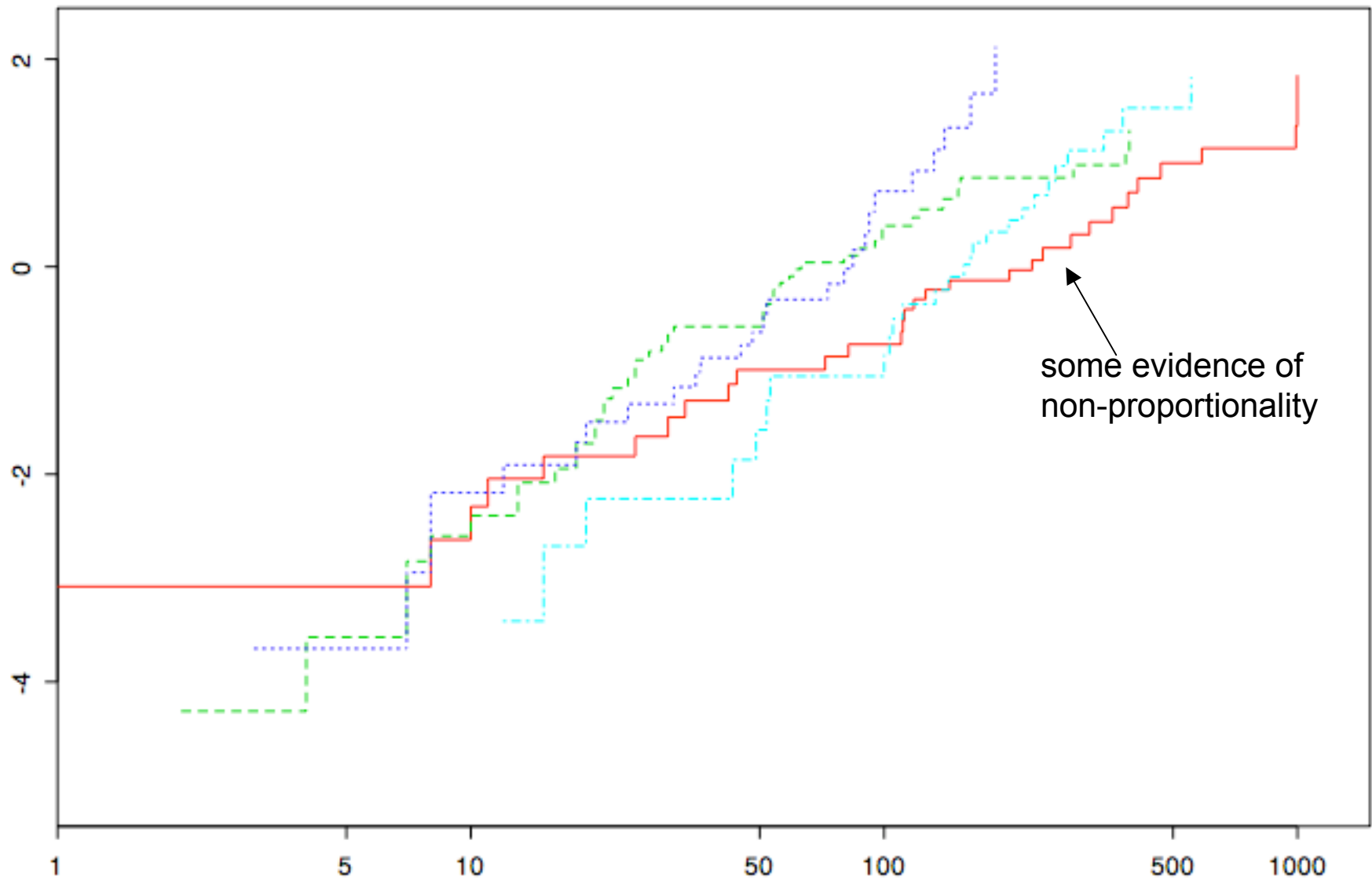
```
plot(survfit(VA.coxs),fun="cloglog", lty=1:4, col=2:5)
legend(300,0.05, c("squamous", "small", "adeno", "large"),
lty=1:4, col=2:5)
```



same thing on a log scale...

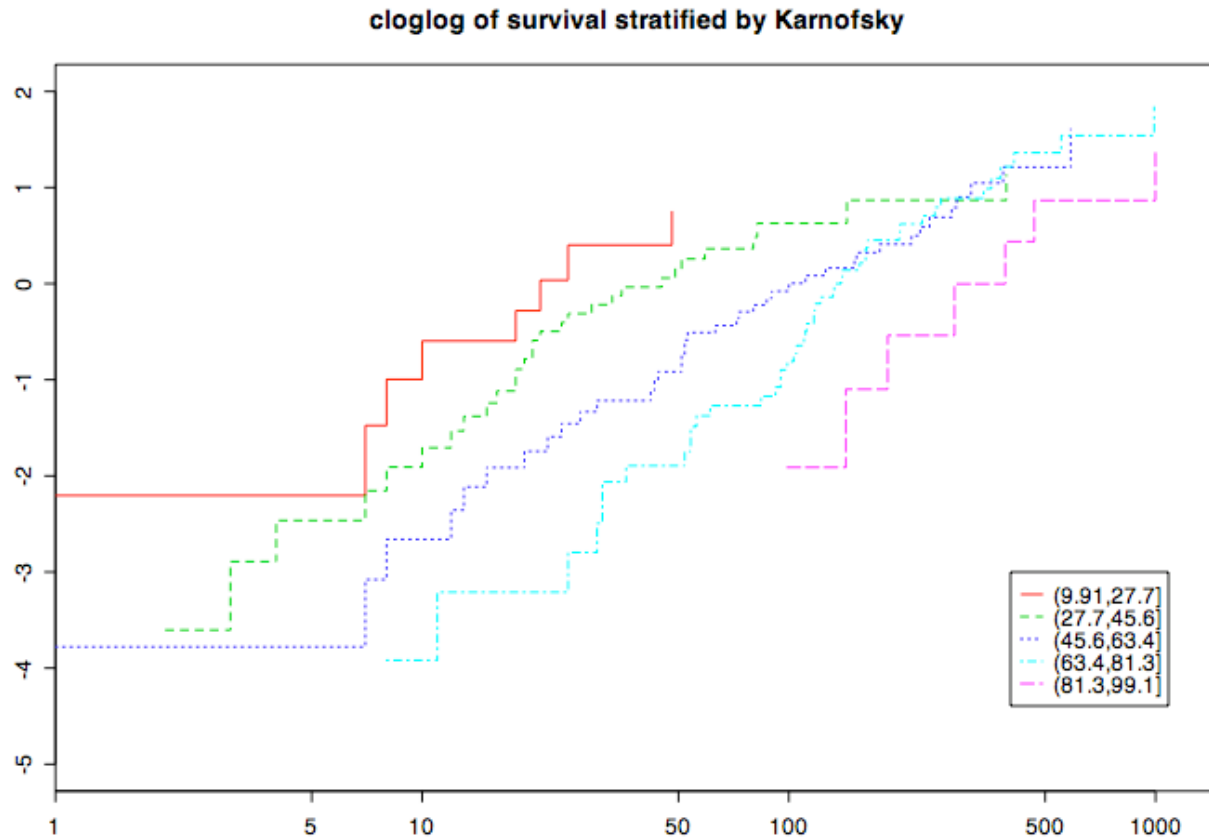


same thing on a log(-log) scale...



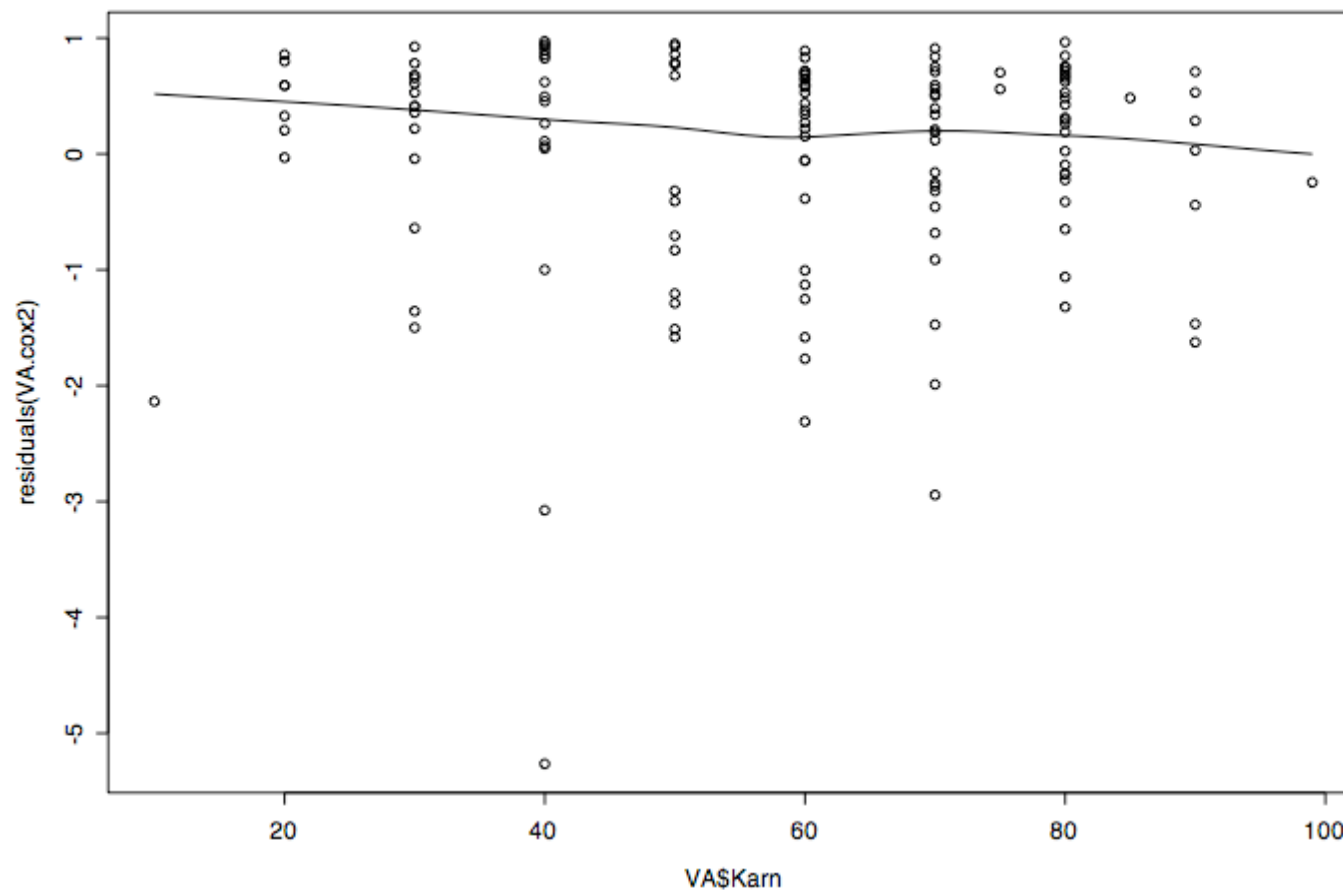
check proportionality for Karnofsky score...

```
cKarn <- factor(cut(VA$Karn, 5))  
VA.cox1 <- coxph(Surv(stime, status) ~  
strata(cKarn)+cell, data=VA)  
plot(survfit(VA.cox1), fun="cloglog")
```



using martingale residuals...

```
VA.cox2 <- coxph(Surv(stime, status) ~ Karn +  
strata(cell), data=VA)  
scatter.smooth(VA$Karn, residuals(VA.cox2))
```



could try parametric - e.g. Weibull

```
VA.wei <- survreg(Surv(stime,status) ~ treat + age + Karn  
+ diag.time + cell + prior, data=VA)
```

```
summary(VA.wei)
```

```
prior10      -0.043898      0.21228 -0.2068 8.36e-01  
Log(scale)   -0.074599      0.06631 -1.1250 2.61e-01  
  
Scale= 0.928  
  
Weibull distribution
```

probably OK to just do exponential:

```
VA.exp <- survreg(Surv(stime,status) ~ treat + age + Karn +  
diag.time + cell + prior, data=VA, dist="exponential")
```

```
> summary(VA.exp)
```

Call:

```
survreg(formula = Surv(stime, status) ~ treat + age + Karn +  
        diag.time + cell + prior, data = VA, dist = "exponential")
```

	Value	Std. Error	z	p
(Intercept)	3.188611	0.70403	4.5291	5.92e-06
treat2	-0.219565	0.19863	-1.1054	2.69e-01
age	0.006108	0.00916	0.6667	5.05e-01
Karn	0.030624	0.00511	5.9958	2.02e-09
diag.time	-0.000297	0.00897	-0.0331	9.74e-01
cell2	-0.820245	0.26211	-3.1294	1.75e-03
cell3	-1.113121	0.27583	-4.0356	5.45e-05
cell4	-0.377220	0.27263	-1.3837	1.66e-01
prior10	-0.049482	0.22687	-0.2181	8.27e-01

Scale fixed at 1

Exponential distribution

Loglik(model)= -716.2 Loglik(intercept only)= -751.2

Chisq= 70.12 on 8 degrees of freedom, p= 4.6e-12

Number of Newton-Raphson Iterations: 5

n= 137

```
VA.cox2 <- stepAIC(VA.cox, ~ .^2)
```